AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listing of claims in the above-referenced application.

Listing of Claims:

1. (Currently Amended) A computer-implemented method for generating a pruned augmented attack tree ~~representing at least one computer attack path in a network~~ comprising:

receiving a starting point of a computer attack with respect to [[said]] a network; [[and]]

inserting a root node into said pruned augmented attack tree for said starting point;

determining whether, for a current node included in said pruned augmented attack tree, to add to said pruned augmented attack tree ~~generating a pruned augmented attack tree representing at least one attack path possible from said starting point, wherein, said starting point is a root of said pruned augmented attack tree, for a current node being evaluated as part of said generating~~, a resulting node and an edge connecting said current node to said resulting node ~~are added to said pruned augmented attack tree~~ if said edge and said resulting node are not already included in said pruned augmented attack tree with said edge connecting an ancestor of the current node to an instance of the resulting node; and

inserting nodes and edges into said augmented pruned attack tree in accordance with said determining, said determining being repeatedly performed for nodes inserted into said pruned augmented attack tree, said inserting being repeatedly performed in accordance with said determining, said pruned augmented attack tree being a pruned version of a full attack tree.

2. (Original) The method of Claim 1, wherein said pruned augmented attack tree is a tree including n levels, said starting point being a root of said tree at level 0, n being at least 0.

3. (Original) The method of Claim 2, wherein a node in said pruned augmented attack tree represents information about at least one of: an attacker state including a host and an attacker access level on said host, and a network state.

4. (Original) The method of Claim 3, wherein an edge from a first node at level x to a second node at level x+1 represents an action while in a first state including a first attacker state corresponding to said first node resulting in a second state including a second attacker state.

5. (Original) The method of Claim 4, wherein said action exploits a vulnerability on a host in said network.

6. (Original) The method of Claim 4, wherein said first attacker state represents a first host and a first attacker access level on said first host, and said second attacker state represents at least one of: a second host and a second attacker access level on said second host, and said first host and a second attacker access level on said first host wherein said second attacker access level represents at least one of: an increase in attacker privilege, an increase in attacker access, and an increase in attacker knowledge.

7. (Original) The method of Claim 1, wherein said current node is at a level n, and said ancestors of said current node are located at levels in said pruned augmented attack tree at a level less than n.

8. (Original) The method of Claim 7, wherein said pruned augmented attack tree is generated using a breadth first search technique in which nodes are added to said pruned augmented attack tree at an nth level prior to adding any node from level n+1 to said pruned augmented attack tree.

9. (Original) The method of Claim 1, wherein a plurality of computer attack paths for said network are represented using a plurality of pruned augmented attack trees, each of said pruned augmented attack trees representing computer attack paths originating from a unique starting point.

10. (Original) The method of Claim 1, wherein said starting point is one of: from within said network and external to said network.

11. (Original) The method of Claim 6, further comprising:

evaluating each action that exploits a vulnerability of a host in accordance with connectivity data.

12. (Currently Amended) The method of Claim 11, wherein said connectivity data, said each action, and said vulnerability are stored in a database and [[determined prior to performing said]] used in connection with generating said pruned augmented attack tree.

13. (Original) The method of Claim 1, wherein, said pruned augmented attack tree has a property that a resulting node at a level "n+1" and an edge connecting a current node at level "n" to said resulting node are included in said pruned augmented attack tree if said edge and said

resulting node are not already included in said pruned augmented attack tree with said edge connecting an ancestor of the current node to an instance of the resulting node, said ancestor being a node at a level "x" < "n", and said instance of the resulting node being at level "x+1".

14. (Original) The method of Claim 1, further comprising:

determining which hosts in said network are equivalent forming a group; and

representing said group with a single host.

15. (Original) The method of Claim 14, wherein a first host is equivalent to a second host if said first and second hosts have a same set of one or more vulnerabilities on a same set of one or more endpoints, said first and second hosts are not administrative hosts, said first and second hosts are not gateways, said first and second hosts have equivalent attack loss values; and said first and second hosts have equivalent connectivity.

16. (Currently Amended) The method of Claim 1, wherein [[said generating uses]] connectivity information is used in connection with generating said pruned augmented attack tree, said connectivity information including a connection between two endpoints representing elements of a configuration of said network.

17. (Original) The method of Claim 16, wherein said connectivity information includes physical connectivity between network interfaces and logical connectivity through network communications protocols.

18. (Original) The method of Claim 16, wherein said connection is associated with a path including one or more hops.

19. (Original) The method of Claim 18, wherein each of said one or more hops is associated with at least one of: a filtering rule, a translation rule, and an interface of a host in said network.

20. (Original) The method of Claim 16, wherein at least one of said endpoints is associated with a vulnerability on said at least one endpoint.

21. (Original) The method of Claim 20, wherein said vulnerability has an associated action resulting in exploitation of said vulnerability.

22. (Original) The method of Claim 21, wherein said associated action is related to an entity representing at least one of: an attacker access level, attacker knowledge level, a change to a network state.

23. (Original) The method of Claim 1, wherein said pruned augmented attack tree is used to determine an effect of preventing at least one action.

24. (Original) The method of Claim 23, further comprising:

modifying said pruned augmented attack tree in accordance with eliminating at least one action in connection with a vulnerability associated with said host producing a modified augmented attack tree; and

evaluating said modified augmented attack tree.

25. (Currently Amended) The method of Claim 1, wherein connectivity data representing connectivity between pairs of endpoints in said network is used [[by said]] in connection with generating said pruned augmented attack tree, and the method further comprising:

automatically generating said connectivity data in accordance with at least one translation rule, at least one filtering rule, and network configuration information.

26. (Original) The method of Claim 25, wherein said at least one translation rule includes at least one of: an address translation rule and a port translation rule.

27. (Original) The method of Claim 25, further comprising:

selecting at least one address of a starting point of a computer attack using at least one rule; and

determining a portion of said connectivity data using said at least one address.

28. (Original) The method of Claims 27, wherein said at least one rule includes at least one of a filtering rule and a translation rule.

29. (Currently Amended) The method of Claim 27, wherein said at least one address is used in connection with [[said]] generating said pruned augmented attack tree to represent an alternate connectivity of a host.

30. (Original) The method of Claim 27, wherein said address is one of an address in accordance with a communications protocol and an address associated with said network.

31. (Original) The method of Claim 5, further comprising:

using vulnerability data to determine at least one of: requirements for an action, an attacker state resulting from an action, and a network state resulting from an action, where said requirements include a locality describing whether a vulnerability can be exploited remotely over a network or locally on a host, said resulting attacker state includes an effect describing an access level or privilege or knowledge after an exploit of a vulnerability, and said resulting network state includes a denial of service describing a loss of service on a host after an exploit of a vulnerability.

32-58. (Cancelled)

59. (Currently Amended) A computer program product for generating a pruned augmented attack tree ~~representing at least one computer attack path in a network~~ comprising executable code that:

receives a starting point of a computer attack with respect to [[said]] a network; [[and]]

inserts a root node into said pruned augmented attack tree for said starting point;

determines whether, for a current node included in said pruned augmented attack tree, to add to said pruned augmented attack tree ~~generates a pruned augmented attack tree representing at least one attack path possible from said starting point, wherein, said starting point is a root of said pruned augmented attack tree, and for a current node being evaluated,~~ a resulting node and an edge connecting said current node to said resulting node ~~are added to said pruned augmented attack tree~~ if said edge and said resulting node are not already included in said pruned augmented attack tree with said edge connecting an ancestor of the current node to an instance of the resulting node; and

inserts nodes and edges into said augmented pruned attack tree in accordance with said determines, repeatedly performing said determines for nodes inserted into said pruned augmented attack tree, repeatedly performing said inserts in accordance with said determines, said pruned augmented attack tree being a pruned version of a full attack tree.

60. (Original) The computer program product of Claim 59, wherein said pruned augmented attack tree is a tree including n levels, said starting point being a root of said tree at level 0, n being at least 0.

61. (Original) The computer program product of Claim 60, wherein a node in said pruned augmented attack tree represents information about at least one of: an attacker state including a host and an attacker access level on said host, and a network state.

62. (Original) The computer program product of Claim 61, wherein an edge from a first node at level x to a second node at level x+1 represents an action while in a first state including a first attacker state corresponding to said first node resulting in a second state including a second attacker state.

63. (Original) The computer program product of Claim 62, wherein said action exploits a vulnerability on a host in said network.

64. (Original) The computer program product of Claim 62, wherein said first attacker state represents a first host and a first attacker access level on said first host, and said second attacker state represents at least one of: a second host and a second attacker access level on said second host, and said first host and a second attacker access level on said first host wherein said second attacker access level represents at least one of: an increase in attacker privilege, an increase in attacker access, and an increase in attacker knowledge.

65. (Original) The computer program product of Claim 59, wherein said current node is at a level n, and said ancestors of said current node are located at levels in said pruned augmented attack tree at a level less than n.

66. (Original) The computer program product of Claim 65, further comprising executable code that generates said pruned augmented attack tree using a breadth first search technique in which nodes are added to said pruned augmented attack tree at an nth level prior to adding any node from level n+1 to said pruned augmented attack tree.

67. (Original) The computer program product of Claim 59, wherein a plurality of computer attack paths for said network are represented using a plurality of pruned augmented attack trees, each of said pruned augmented attack trees representing computer attack paths originating from a unique starting point.

68. (Original) The computer program product of Claim 59, wherein said starting point is one of: from within said network and external to said network.

69. (Original) The computer program product of Claim 64, further comprising:

executable code that evaluates each action that exploits a vulnerability of a host in accordance with connectivity data.

70. (Original) The computer program product of Claim 69, further comprising executable code that stores said connectivity data, said each action, and said vulnerability in a database prior to generating said pruned augmented attack tree.

71. (Original) The computer program product of Claim 59, wherein, said pruned augmented attack tree has a property that a resulting node at a level "n+1" and an edge connecting a current node at level "n" to said resulting node are included in said pruned

augmented attack tree if said edge and said resulting node are not already included in said pruned augmented attack tree with said edge connecting an ancestor of the current node to an instance of the resulting node, said ancestor being a node at a level "x" < "n", and said instance of the resulting node being at level "x+1".

72. (Original) The computer program product of Claim 59, further comprising executable code that:

determines which hosts in said network are equivalent forming a group; and

represents said group with a single host.

73. (Original) The computer program product of Claim 72, wherein a first host is equivalent to a second host if said first and second hosts have a same set of one or more vulnerabilities on a same set of one or more endpoints, said first and second hosts are not administrative hosts, said first and second hosts are not gateways, said first and second hosts have equivalent attack loss values; and said first and second hosts have equivalent connectivity.

74. (Currently Amended) The computer program product of Claim 59, wherein ~~said executable code that generates said pruned augmented attack tree uses~~ connectivity information is used in connection with generating said pruned augmented attack tree, said connectivity information including a connection between two endpoints representing elements of a configuration of said network.

75. (Original) The computer program product of Claim 74, wherein said connectivity information includes physical connectivity between network interfaces and logical connectivity through network communications protocols.

76. (Original) The computer program product of Claim 74, wherein said connection is associated with a path including one or more hops.

77. (Original) The computer program product of Claim 76, wherein each of said one or more hops is associated with at least one of: a filtering rule, a translation rule, and an interface of a host in said network.

78. (Original) The computer program product of Claim 74, wherein at least one of said endpoints is associated with a vulnerability on said at least one endpoint.

79. (Original) The computer program product of Claim 78, wherein said vulnerability has an associated action resulting in exploitation of said vulnerability.

80. (Original) The computer program product of Claim 79, wherein said associated action is related to an entity representing at least one of: an attacker access level, attacker knowledge level, a change to a network state.

81. (Original) The computer program product of Claim 59, further comprising executable code that uses said pruned augmented attack tree to determine an effect of preventing at least one action.

82. (Original) The computer program product of Claim 81, further comprising executable code that:

modifies said pruned augmented attack tree in accordance with eliminating at least one action in connection with a vulnerability associated with said host producing a modified augmented attack tree; and

evaluates said modified augmented attack tree.

83. (Currently Amended) The computer program product of Claim 59, wherein connectivity data representing connectivity between pairs of endpoints in said network is used in connection with generating said pruned augmented attack tree ~~by said executable code that generates~~, and the computer program product further comprising executable code that:

automatically generates said connectivity data in accordance with at least one translation rule, at least one filtering rule, and network configuration information.

84. (Original) The computer program product of Claim 83, wherein said at least one translation rule includes at least one of: an address translation rule and a port translation rule.

85. (Original) The computer program product of Claim 83, further comprising executable code that:

selects at least one address of a starting point of a computer attack using at least one rule; and

determines a portion of said connectivity data using said at least one address.

86. (Original) The computer program product of Claim 85, wherein said at least one rule includes at least one of a filtering rule and a translation rule.

87. (Currently Amended) The computer program product of Claim 85, wherein said at least one address is used in connection with [[said]] generating said pruned augmented attack tree to represent an alternate connectivity of a host.

88. (Original) The computer program product of Claim 85, wherein said address is one of an address in accordance with a communications protocol and an address associated with said network.

89. (Original) The computer program product of Claim 63, further comprising:

executable code that uses vulnerability data to determine at least one of: requirements for an action, an attacker state resulting from an action, and a network state resulting from an action, where said requirements include a locality describing whether a vulnerability can be exploited remotely over a network or locally on a host, said resulting attacker state includes an effect describing an access level or privilege or knowledge after an exploit of a vulnerability, and said resulting network state includes a denial of service describing a loss of service on a host after an exploit of a vulnerability.

90-116. (Cancelled)